NIT-226
NT0157US

Title of the Invention

DATABASE PROCESSING SYSTEM, METHOD, PROGRAM
AND PROGRAM STORAGE DEVICE

Inventors

Koji YANASE
Seiichi MANIWA

DATABASE PROCESSING SYSTEM, METHOD, PROGRAM AND PROGRAM STORAGE DEVICE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a database processing method.

Prior Art

As a data storage method of table of relational database management system, there is a partitioning of table. As a method of table partitioning storage, there are known methods such as the key range partitioning, and hash partitioning. A technique applying these techniques is disclosed in the JP-A No. H6-139119 and JP-A No. H6-314299. In addition, a partitioning method that combines a plurality of steps of these methods is disclosed in the JP-A No. H10-269225.

The reason for partitioning these tables is not simply because a large amount of data may not be stored in one external storage device. The improvement of processing speed and concurrency by referring and updating only the specific external storage device or a logical database storage area at the time of searching is the aim, also the merits in the operation side by allowing localizing the backup and reconstruction is another aim.

In the key range partitioning and the hash partitioning, there are cases in which values of one column of table are used and cases in which values of a plurality of columns of table are used. When values of a plurality of columns of table are used, values of a plurality of columns are concatenated to insistently determine a logical database storage area as one key. In addition, in the partitioning method that combines a plurality of steps of these methods such as that disclosed in the above JP-A No. H10-269225, the multiple components of hardware configuration is used by such as the partitioning of database of first step to each of computers in a group of computers, the partitioning of database of second step to each of processors in each computer, furthermore the partitioning of database of third step to each of external storage device in each of processors. The partitioning in each phase in this is to insistently use one key.

As another conventional example, there is JP-A No. H10-240744. This is a partitioning of database with a range of one key. Furthermore, when defining a table the storage position of data is not itself determinable from the value of key. When storing data, in order to memorize one by one where is stored the data, the number of storage range of records and the key value that is included in the records are corresponded to store in the key information storage area.

Still another conventional example, there is JP-A No. H5-334165. In this disclosure the key that is used for the key range partitioning is only one key of primary key. Then, the partitioned part of table is distributed to the local database processing means to maintain. In such a circumstance the searching by a second key is enabled.

In the partitioning storage techniques of the Prior Art, when one table is partitioned to a plurality of storage areas in one processor, the key that is used for the partitioning condition is only one. Therefore in the condition that is other than the partitioning key that is only one, the narrowing of storage area is not allowed, so that the every data storage area is to be subject to be processed. Also, when a plurality of keys is used in a combination of multiple steps, the system configuration of hardware is dependent.

SUMMARY OF THE INVENTION

The present invention has been made in view of the above circumstances and has an object to overcome the above problems and to provide a database management system that solves the above problems.

In order to achieve the object as have been described above, by specifying the partitioning condition of each by using N keys where N > 1 from the columns that constitute one table of a relational database, a partitioning table is defined that has a data storage areas that is a combination of every partitioned unit by each of keys to be N dimensions. In this case the hash partitioning may be combined together therewith. At this time the partitioned definition information is stored in a dictionary of the same layer, that is, in the same Data Base Management system (DBMS). In accordance with this partitioned definition information, data is stored in a plurality of logical database storage areas of multiple dimensions, at the request of query by determining the subminimal area of logical database storage from every value of partitioning keys to provide a database management system that achieves high-speed database accesses.

Additional objects and advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram of database management system in accordance with the present invention;

Fig. 2 is an example and image diagram of definition statement of multiple dimension partitioning;

Fig. 3 is an example of the contents of dictionary table that stores the definition information of partitioned table;

Fig. 4 is an example of the contents of dictionary table that stores the definition information of the storage area;

Fig. 5 is a schematic diagram illustrating the configuration with the functional components of the database management system in accordance with the present invention;

Fig. 6 is a schematic diagram illustrating the flow in the table definition;

Fig. 7 is a schematic diagram illustrating the flow at the time of data insertion;

Fig. 8 is a schematic diagram illustrating the flow of process that specifies the storage area; and

Fig. 9 is a schematic diagram illustrating the flow at the time of data search.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is to allow search from the viewpoint of dynamics or multiple dimension, hereinafter one preferred embodiment of the present invention will be described in greater details with reference to accompanying figures of Fig. 1 to Fig. 9.

A structure of a database management system that implements the present invention is shown in Fig. 1. The database management system is constituted of one processor or a plurality of processors 102 that are connected by a high speed network or an inter processor connecting apparatus 101, a plurality of external storage apparatuses 103 in each of processors are also components. A logical database storage area 104 is allocated on each of external storage apparatuses, there may be cases in which it may be allocated among a plurality of external storage apparatuses. A dictionary 105 that stores the definition information of tables and the storage areas is present on the external storage apparatus, there may be cases in which it is controlled by a proprietary processor and there may be cases in which it is controlled by sharing a plurality of processors.

A dictionary table 110 of a partitioning key definition information management table is a table that manages partition definition information that is constituted of a column of table name 111, a column of partitioning key ID 112 (unique value in one table that may correspond to the specified order at the time of table definition), a column of the name of column that is a member of the partitioning key 113, a column of data types of the columns 114, and a column of the number of key ranges 115. A B tree index that is constituted of the columns 111 and 112 of this table is present in the same storage area as the dictionary table to be used for accelerating the searching process of the partitioning key definition information by the table name. The dictionary table 110 is corresponding to the D301 of Fig. 3.

A dictionary table 120 is a key range information management table that manages the range information of the Partitioning Key that is constituted of a column of table name 121, a column of partitioning key IDs 122, a column of boundary values 123 that is the upper limit of each of the partitioning areas, a column of partitioning key range numbers 124 (that become unique in the same partitioning key) that are allocated from one in accordance with the ascending order of the boundary values in the same partitioning key. Also, a B tree index that is constituted of the columns 121, 122 and 124 of this table is present in the same storage area as the dictionary table to be used for accelerating the searching process of the key range information by the table name and the partitioning key IDs. The dictionary table 120 is corresponding to the D302 of Fig. 3.

A dictionary table 130 is a partitioned table storage area management table that is constituted of a column of table name 131, a column of storage area order number allocated in the ascending order from one in accordance with the order specified by the table definition statement for each storage area (unique number in one table) 132, and a column of storage area name 133. In addition, a B tree index that is constituted of the columns 131 and 132 of this table is present in the same storage area as the dictionary table to be used for accelerating the searching process of the storage area name by the table name or by the table name and the storage area order number. The dictionary table 130 is corresponding to the D303 of Fig. 3.

A dictionary table 140 is a storage area definition information management table that manages the definition information of every storage areas in the system that is constituted of a column of storage area name 141, a column of file name of external storage device in which the storage area is present 143, and a column of processor name 142 that the external storage device is managed. In addition, a B tree index that is constituted of the column 141 of this table is present in the same storage area as the dictionary table to be used for accelerating the searching process of the storage area definition information by the storage area name. The dictionary table 140 is corresponding to the D401 of Fig. 4.

One example of table definition statement of SQL that represent the characteristics of the present invention and the partitioned image is shown in Fig. 2. In the table of sale achievement shown in the example, among a number of component columns, three columns (registration day, branch code number, and goods classification) are independent partitioning keys respectively, for each key two partitioning boundary values are specified so as to be partitioned into three key ranges, so that in this example 27 (3 by 3 by 3) storage areas are partitioned. However, in the present invention, the number of partitioning keys, and the partitioning number by one partitioning key have no upper limit in particular, in practice it is possible to be partitioned into a huge number of storage areas. Each storage area may be present on which external storage device of which processor. It can be positioned freely in accordance with the data capacity and access frequency in the combination of each of partitioned range. Off course it can be closed to one processor.

The contents of dictionary table that manages the partitioned definition information of the table of the case when the table of Fig. 2 is defined is shown in Fig. 3. The D301 that is corresponding to the dictionary table 110 is a table that manages the definition information of partitioning key, that stores for each partitioning key the ID thereof (the specified order at the time when the table definition), the component columns (that is a member of partitioning key) and its data type (the number of bytes of the character data is indicated in the parenthesis), the number of partition by the boundary values and the like. The D302 that is corresponding to the dictionary table 120 is a table that manages the range information of partitioning keys, which stores the boundary values that become the upper limit of each of partitioned range in each of partitioning keys, and the key range number that is allocated in the ascending order from one. The D303 that is corresponding to the dictionary table 130 is a table that manages the information of storage area that stores the data of insertion, that stores the storage area order number that is allocated from one in the ascending order in accordance with the sequence specified in the table definition statement for each of storage area. In the example as have been described above, each of three partitioning keys is partitioned to three areas so that the storage area will be partitioned from 1 to 27.

An example of the contents of the dictionary table that manages the definition information of the storage area is shown in Fig. 4. The D401 that is corresponding to the dictionary table 140 is a table that manages the definition information of every storage areas that are defined in the database management system, that stores the information of management node (processor) and external storage device for each of storage area.

The schematic diagram of functional structure of the database management system that implements the present invention is shown in Fig. 5. The database management system is in the present invention implemented by a program, and is possible to be recordable on a computer readable recording medium. The database management system is constituted of a command analyzer 501, which receives an SQL, an access path generator (optimizer) 502, which generates an execution procedure of SQL, an SQL execution controller 503 that performs data processing in accordance with the execution procedure, a dictionary manager 504 that manages a dictionary, a storage area specification component 505 that specifies the storage area subject to be accessed in accordance with the value of input data and the search condition, a database I/O handler 506 that manages the I/O of data, a communication controller 507 that controls the communication with other processors. There are either cases in which the functional components 501 through 506 are incorporated to every processor or cases in which the functional components incorporated may be different from one processor to another in accordance with the roles of processors, the communication controller 507 is incorporated to all of the processors only in case of a plurality of processor configurations. The communication controller 507 may be incorporated where necessary.

Fig. 2 is used by way of example for describing about the method of table definition of the table "sale achievement" of the relational database. The relational database user specifies a plurality of partitioning keys, and specifies a boundary value for each of partitioning keys respectively. In this example first partitioning key is registration day, second partitioning key is branch code number, third partitioning key is goods classification, and two boundary values for each are specified (resulting in tree partitioned ranges for each). As shown in Fig. 2, these storage areas are specified by one SQL table definition statement. The storage areas that may satisfy all of the combination in accordance with the order (rule) may be specified such as the storage area (A111) in which the data that satisfies the first range of the first partitioning key and the first range of the second partitioning key and the first range of the third partitioning key is stored, the storage area (A112) in which the data that satisfies the first range of the first partitioning key and the first range of the second partitioning key and the second range of the third partitioning key is stored, the storage area (A113) in which the data that satisfies the first range of the first partitioning key and the first range of the second partitioning key and the third range of the third partitioning key is stored, the storage area (A121) in which the data that satisfies the first range of the first partitioning key and the second range of the second partitioning key and the first range of the third partitioning key is stored, and so on.

Fig. 6 is shown with respect to the flow of the table definition. The command analyzer 501 analyzes a table definition statement (601). The analyzed partition definition information is passed to the dictionary manager 504 to store in a dictionary table as shown in Fig. 1. For the definition information for each of partitioning keys, Dictionary Manager determines Partitioning Key IDs in correspondence with the destination sequence of each Partitioning Key. Then the rows, each row is constituted of the Partitioning Key ID, name of partitioned table, name of column that is a member of the Partitioning Key (one key may be constituted of a plurality of columns), data type of the column, and the number of key ranges, are inserted into the dictionary table 110 (602). In the example shown in Fig. 2, three columns of data will be registered to the D301. With respect to the definition information of each of partition range in each of respective partitioning keys, Dictionary Manager determines Key range number according to the ascending of boundary values for each Key range. Then the rows, each row is constituted of the Key range number, name of partitioned table, Partitioning Key ID, and boundary value, are inserted into the dictionary table 120 (603). In the example shown in Fig. 2, 9 rows of data are registered to the D302 because each of respective three partitioning keys is partitioned to three partitioned ranges. The storage area definition information with respect to the table definition, Dictionary manager determines storage area order number in correspondence with the destination sequence of each storage area. Then the rows, each row is constituted of the storage area order number, name of partitioned table, storage area name, are inserted into the dictionary table 130 (604). The number of rows corresponds to the number of storage areas, in the example as shown in Fig. 2 27 rows of data are registered to the D303.

The flow in case of insertion of data to the sale achievement is shown in Fig. 7. When a database user issues an SQL that is indicative of an insertion of data from a terminal, the command analyzer 501 receives SQL and analyzes the SQL to pass the control to the access path generator (optimizer) 502 that may generate the optimum execution sequence of SQL (701). The access path generator (optimizer) 502 passes the control to the storage area specification component 505 together with the partitioning key value of the data to be inserted (702).

The storage area specification component 505 determines the partitioning definition information (110, 120) of the target table that is stored in the dictionary through the dictionary manager 504 (703), then specifies a storage area from the partitioning definition information determined and the partitioning key value of the insertion data (704, will be detailed later). The access path generator (optimizer) 502 generates access path on the basis of information of a storage area specified in the storage area specification component, then passes the control to the SQL execution controller 503 (705). The SQL execution controller 503 transfer control to database I/O handler 506 of the same processor or that 506 of other processors through communication controller 507 according to the access path (706). The database I/O handler 506 inserts the value of the rows into the specified storage area (707).

The flow of the storage area specification processing is shown in Fig. 8. In the figure the key range number for each of partitioning keys is determined from values of the insertion data, in addition thereto, by referring to the above example, which order number of storage areas from 1 to 27 is equivalent to the storage area is determined. At this point, when data is inserted, always a value is given with respect to the insertion data for each partitioning key. At first, the initial value 1 is assigned to the variable n that stores the partitioning key ID (801). (That is, processing the first partitioning key; with reference to the above example, processing the "registration day") Next, the all bits of the element S(n) of a bit string variable array S are set off. The length of each element is equal or greater than the number of storage areas of the Partitioned Table(bits), Because the position of each bit is equivalent to a storage area order number (802) (that is, in the above example, 27 bit strings that corresponds to the partitioning key "registration day" are provided and set to '0' as the initial value). Then in case of the specification process at the time of searching, since there are cases in which no condition value is specified with respect to a partitioning key, if the search condition about the Partitioning key (ID is n) is not specified in the SELECT statement, all bits of S(n) are set ON, and go to 811 (all storage area for the candidates of searching) (803). In case of insertion or in case in which the search condition is specified with respect to a partitioning key that the partitioning key ID be n, all the numbers (already obtained from the partitioning key definition information management table 110) of Key ranges of Partitioning key (ID is greater than n) are multiplied. Then the value (if the Partitioning key ID that is greater than n is not present, the value is 1) is substituted for variable C (804). (in the above example, if n = 1, that is, if the processing of "registration day" is done, then 9, that is obtained by multiplying the number of partitioning 3 of branch code number of the partitioning key ID 2 with the number of partitioning 3 of "Goods classification" of the partitioning key ID 3, is substituted for the variable C). Then the number of partitioning of the partitioning key that the partitioning key ID is n will be substituted for the variable D (805) (in case of "registration day" D = 3).

Then the value of partitioning key that the partitioning key ID is n (the insertion value or searching condition) is compared with the boundary value that is sorted in the ascending order of partitioning keys obtained from the dictionary table 120 (searching by binary search so as to accelerate the processing even if the number of partition ranges is larger), to determine the partitioning key range number, then the key range number is substituted for variable F (806). Then (F - 1) x C + 1 is substituted for variable G (807) (if the partitioning key range number 2 is F, then G = 10). Then in variable S(n), C bits that are continuous from the Gth bit are set ON (808) (in the above example, 9 bits that are continuous from the 10th bit of S(1) are set to ON).

Then C x D is added so as to update the value of G to the next specified storage area order number (809). If G is not greater than the number of storage areas, go to 808 to repeat the processing (810). If otherwise the number of G is greater than the maximum value of the storage area order number (in the above example if beyond 27, the processing about that partitioning key has been already completed), n is added to 1 in order to perform specification processing on the basis of the next partitioning key (811). If n is not greater than the number of partitioning keys, go to 802 to repeat processing (810) (that is, to perform the same processing as above with respect to the "branch code number" of the partitioning key ID 2 and the "goods classification" of the partitioning key ID 3). If n is greater than the number of partitioning keys, the And of variables (from S(1) to S(n-1)) is calculated (in the above example, 27 bit strings), and Storage area order numbers are determined from the result (in the above example, among 27 bits of each of respective partitioning keys, the common position of ON bits is determined. That is the storage area order number of the insertion data or selection data). The storage area order number corresponding to the bit position of ON (there is always one at the time of insertion) and table name are used for search the dictionary table 130 to determine a specified storage area name. Then information of processor name and external storage device that includes the storage area are selected from the dictionary table 140 by using the storage area name as search condition. At the time of data insertion only one storage area is determined from all of the partitioning key values (813).

The flow in case of searching data from the sale achievement table is shown in Fig. 9. Once a database user issues SQL that is indicative of data searching from his terminal, the command analyzer 501 receives SQL(SELECT) and analyzes the SQL. The control is passed to the access path generator (optimizer) 502 that generates the optimum execution sequence of SQL (901). Access path generator (Optimizer) 502 transfers the values (or the range) of the Partitioning keys that are specified in search condition to storage area specification component 505 together with the control (902). The Storage area specification component 505 receives the definition information (110, 120) of the partitioned table (that is the object of select) from Dictionary manager 504 that retrieves the dictionary tables (903), to specify a storage area from thus determined partitioning definition information and the partitioning key value of the search condition (904, details is described the above) (at this time, if a value of search condition is not given to a partitioning key, a search about a plurality of storage areas is to be performed). The access path generator (optimizer) 502 generates access path on the basis of information of a storage area specified in the storage area specification component (905) and passes the control to the SQL execution controller 503. The SQL execution controller 503 then transfer control to database I/O handler 506 of the same processor or that 506 of other processors through communication controller 507 according to the access path (906). The database I/O handler accesses to the specified storage area to retrieve data (907). At the time of search of data, only if the search condition with respect to all of partitioning keys is specified only one storage area is determined. However, if the search condition with respect to any one of partitioning keys is specified, the storage area may be narrowed to eliminate the storage areas that are not required to be accessed from the search object.

The present invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. For instance, as have been described above, in accordance with the present invention, regardless of presence or absence of the hierarchical structure such as by partitioning by each computer (system), partitioning by each processor in each computer (system), or partitioning by each storage device in each processor, by specifying a plurality (N) of partitioning keys, a large amount of storage data may be allowed to be partitioned and stored in the storage areas of N dimension. In addition, because the search processing is performed only in the optimum least required storage areas even when any partitioning key value is specified, the concurrent effectiveness and the throughput may be improved. In addition, the execution unit of backups and data reconstruction also may be set in a more flexible manner so that the operability is helped to be improved. Furthermore, a grasp of data as such may be seen from a variety of aspects by changing the key items and the numbers. From a user's view, the data items of the data to be analyzed may be allowed to change, as well as the response may be improved due to the increased throughput, the multidirectional analytical results may be obtained faster than ever.

It is to be understood that the present invention is not to be limited to the details herein given but may be modified within the scope of the appended claims.